

1 **IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

2 Application Serial No.09/878,950
3 Filing Date06/11/2001
4 Inventorship Sanders et al.
5 Assignee.....Microsoft Corporation
6 Group Art Unit 2194
7 Examiner Wu, Qing Yuan
8 Attorney's Docket No.MS1-771US
9 Title: Methods and Arrangements for Routing Server Requests to Worker Processes
10 Based on URL

11 **APPEAL BRIEF**

12 To: MS: Appeal Brief - Patents
13 Commissioner for Patents
14 P.O. Box 1450
15 Alexandria, VA 22313-1450

16 From: Tim R. Wyckoff (Tel. 206.315.4001 x110; Fax 206.315.4004)

17 **Customer No. 22801**

18 Lee & Hayes PLLC
19 421 W Riverside Avenue, Suite 500
20 Spokane, WA 99201

21 **INTRODUCTORY COMMENTS**

22 Pursuant to 37 C.F.R. § 41.37, Appellant hereby submits an Appeal Brief
23 for Application Serial No. 09/878,950 filed June 11, 2001. A Notice of Appeal
24 was filed on December 13, 2005. Accordingly, Appellant appeals to the Board of
25 Patent Appeals and Interferences (hereinafter "Board") seeking review of the
 Office's rejections.

TABLE OF CONTENTS

<u>Appeal Brief Items</u>	<u>Page</u>
(i) Real Party in Interest	4
(ii) Related Appeals and Interferences	4
(iii) Status of Claims	4
(iv) Status of Amendments	5
(v) Summary of Claimed Subject Matter	6
(vi) Grounds of Rejection to be Reviewed on Appeal	8
(vii) Argument	9
(A) Claim 6 fully conforms with at least 35 U.S.C. § 112, second paragraph	9
(B) Claims 1-11 and 13-24 are allowable because the combination of <i>AAPA</i> and <i>Benitez</i> does not disclose or suggest the subject matter of these claims	10
(C) Claims 1-11 and 13-24 are allowable because the Office has used impermissible hindsight as motivation to combine <i>AAPA</i> and <i>Benitez</i>	22
(D) Claims 1-11 and 13-24 are allowable because <i>Benitez</i> is from non-analogous art	25
(viii) Claims Appendix	29

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

TABLE OF CONTENTS CONTINUED

Appeal Brief Items

Page

(ix) Evidence Appendix

37

(x) Related Proceedings Appendix

38

1 **(i) Real Party in Interest**

2 The real party in interest is the Microsoft Corporation, the assignee of all
3 right and title to the subject invention.

4
5 **(ii) Related Appeals and Interferences**

6 Appellant is not aware of any other appeals or interferences which will
7 directly affect, be directly affected by, or otherwise have a bearing on the Board's
8 decision to this pending Appeal.

9
10 **(iii) Status of Claims**

11 Allowed Claims: No claims have been allowed.

12 Canceled Claims: Claims 12 and 26-33 were previously canceled.

13 Originally Presented Claims: Claims 1-33 were originally presented
14 when this Application was filed.

15 Pending Claims: Claims 1-11 and 13-24 stand rejected and are pending
16 in this Application as set forth in the Claims Appendix on page 29.

17 Appealed Claims: All of the pending claims are subject to this Appeal.
18 Claim 6 stands rejected under 35 U.S.C. § 112, second paragraph, as being
19 indefinite for failing to particularly point out and distinctly claim the subject
20 matter which the Appellant regards as the invention. Claims 1-11 and 13-33 are
21 rejected under 35 U.S.C. § 103(a) as being unpatentable over "Applicant Admitted
22 Prior Art", in view of Benitez et al. Both these rejections are set forth in the Final
23 Office Action dated June 13, 2005.

1 The Final Office Action also set forth a rejection of claims 1-11 under 35
2 U.S.C. § 101. This rejection was withdrawn by the Office in an Advisory Action
3 dated September 7, 2005.

4
5 **(iv) Status of Amendments**

6 A Final Office Action was issued on June 13, 2005 whereupon Appellant
7 filed a Response on August 12, 2005 to address the rejections of pending claims.
8 Claim 6 was amended in the Response in an effort to overcome the rejection under
9 35 U.S.C. § 112, second paragraph. No claims were canceled in the Response.

10 An Advisory Action was issued on September 7, 2005 in response to the
11 Response filed on August 12, 2005. The Advisory Action stated that the
12 rejections under 35 U.S.C. §§ 112 and 103(a) were not overcome. However, the
13 Advisory Action did indicate that the rejection under 35 U.S.C. § 101 was
14 overcome.

15 The Appellant filed another Response on September 13, 2005 that canceled
16 claims 25-33 and amended claim 6 again in an effort to overcome the rejection
17 under 35 U.S.C. § 112, second paragraph. The Response asked the Office to
18 reconsider the outstanding rejections.

19 After an unexplained delay, a second Advisory Action was issued on
20 February 7, 2006 in response to the Response filed on September 13, 2005. The
21 Appellant notes that the Advisory Action was mailed almost five months after the
22 Response of September 13, 2005 was submitted. The inappropriate delay
23 necessitated the filing of a Notice of Appeal on December 13, 2005 to assure the
24 present Application did not go Abandoned. The Advisory Action stated that the
25

1 rejection under 35 U.S.C. § 103(a) was not overcome. The Advisory Action did
2 not address the rejection under 35 U.S.C. § 112, second paragraph.

3 The Advisory Actions of September 7, 2005 and February 7, 2006 each
4 indicated that the Responses filed after the Final Office Action dated June 13, 2005
5 would be entered.

6
7 **(v) Summary of Claimed Subject Matter**

8 The following is a concise explanation of each independent claim 1 and 13
9 involved in the Appeal and includes references to the specification by page and
10 line number, and to the drawings. The independent claims are not to be limited
11 solely to the elements identified by the reference characters.

12 In the following, a concise explanation of each dependent claim 3, 14 and
13 15 involved in the Appeal is also presented. The concise explanation includes
14 references to the specification by page and line number, and to the drawings. The
15 dependent claims are not to be limited solely to the elements identified by the
16 reference characters.

17 **Claim 1** recites a method comprising: receiving a request from a client
18 device (*page 17, lines 23-24 and figure 6, block 604*), the request comprising a
19 hierarchical identifier (*page 18, lines 1-2 and figure 6, block 606*); comparing the
20 hierarchical identifier (*pages 15-16 and figure 5 discuss/illustrate hierarchical*
21 *identifiers in detail*) with at least a portion of a configuration file (*configuration*
22 *file 404, page 14, lines 3-11*) to identify an appropriate user-mode process (*page*
23 *14, lines 3-19, web server process 408, worker process 410, and figure 4*) for
24 handling the request (*page 18, lines 3-9 and figure 6, block 606*); and providing
25

1 the request to the identified appropriate user-mode process (*page 18, lines 10-14*
2 *and figure 6, block 608*).

3 **Claim 3** recites the method as recited in Claim 2, wherein generating the
4 configuration file (*configuration file 404, page 14, lines 3-11*) comprises: defining
5 one or more logical associations between at least one candidate hierarchical
6 identifier (*page 15, lines 17-22 and figure 5*) and at least one candidate user-mode
7 process (*page 15, lines 17-22 and figure 5*); and maintaining the one or more
8 logical associations in a configuration store (*configuration store 414, page 14, line*
9 *25*).

10 **Claim 6** recites a method as recited in Claim 1, further comprising
11 selectively queuing the request prior to providing the request to the identified most
12 applicable user-mode process (*page 14, lines 3-19, web server process 408,*
13 *worker process 410, and figure 4*).

14 **Claim 13** recites a computer-readable medium having computer-executable
15 instructions for performing steps comprising: causing a kernel-mode process in a
16 server device (*page 13, lines 19-23, web server arrangement 400*) to compare a
17 hierarchical identifier (*pages 15-16 and figure 5 discuss/illustrate hierarchical*
18 *identifiers in detail*) associated with a client device generated request with at least
19 a portion of a configuration file (*configuration file 404, page 14, lines 3-11*) to
20 identify a most applicable user-mode process (*page 14, lines 3-19, web server*
21 *process 408, worker process 410, and figure 4*) for handling the client device
22 generated request within the server device (*page 18, lines 3-9 and figure 6, block*
23 *606*); and causing the kernel-mode process to provide the client device generated
24 request to the identified most applicable user-mode process (*page 18, lines 10-14*
25 *and figure 6, block 608*).

1 **Claim 14** recites the computer-readable medium as recited in Claim 13,
2 having further computer-executable instructions for performing steps comprising:
3 causing a user-mode administrative process (*user-mode Web Administration*
4 *Service (WAS) process 412, page 14, lines 23-24*) to generate the configuration file
5 (*configuration file 404, page 14, lines 3-11*).

6 **Claim 15** recites the computer-readable medium as recited in Claim 14,
7 wherein causing the user-mode administrative process (*user-mode Web*
8 *Administration Service (WAS) process 412, page 14, lines 23-24*) to generate the
9 configuration file (*configuration file 404, page 14, lines 3-11*), further includes:
10 providing a configuration store (*configuration store 414, page 14, line 25*) suitable
11 for access by the user-mode administrative process (*user-mode Web*
12 *Administration Service (WAS) process 412, page 14, lines 23-24*), wherein the
13 configuration store (*configuration store 414, page 14, line 25*) defines one or more
14 logical associations between at least one candidate hierarchical identifier (*pages*
15 *15-16 and figure 5 discuss/illustrate hierarchical identifiers in detail*) and at least
16 one candidate user-mode process (*page 14, lines 3-19, web server process 408,*
17 *worker process 410, and figure 4*).

18
19 **(vi) Grounds of Rejection to be Reviewed on Appeal**

20 *Claim Rejection Under 35 U.S.C. § 112*

21 Claim 6 stands rejected under 35 U.S.C. § 112, second paragraph, as being
22 indefinite for failing to particularly point out and distinctly claim the subject
23 matter which the Appellant regards as the invention.
24
25

1 Claim Rejection Under 35 U.S.C. § 103

2 Claims 1-11 and 13-33 are rejected under 35 U.S.C. § 103(a) as being
3 unpatentable over “Applicant Admitted Prior Art” (hereafter, “*AAPA*”), in view of
4 Benitez et al (U.S. PG Pub 20020161908A1) (hereafter “*Benitez*”).

5
6 **(vii) Argument**

7
8 **(A) Claim 6 fully conforms with at least 35 U.S.C. § 112, second**
9 **paragraph**

10
11 **Claim 6**

12 Claim 6 recites a method as recited in Claim 1,

- 13
14
 - further comprising selectively queuing the request prior to providing
the request to the identified most applicable user-mode process.

15
16 Appellant amended claim 6 to rectify a minor antecedent basis informality
17 therein. The Appellant submits that amended claim 6 recites subject matter that
18 has proper antecedent basis.

19 For the reasons given above, the Board is asked to reconsider and withdraw
20 the 35 U.S.C. § 112 rejection.

(B) Claims 1-11 and 13-24 are allowable because the combination of *AAPA* and *Benitez* does not disclose or suggest the subject matter of these claims

Claims 1-2 and 5-12

Claim 1 recites a method comprising:

- receiving a request from a client device, the request comprising a hierarchical identifier;
- comparing the hierarchical identifier with at least a portion of a configuration file to identify an appropriate user-mode process for handling the request; and
- providing the request to the identified appropriate user-mode process.

An implementation described in the present Application teaches receiving a request at a *kernel-mode* universal listener (UL) service 402 and comparing the request with content of a *kernel-mode* configuration file 404. (*See page 14, lines 3-5; and Fig. 4 of the present Application.*) Hierarchical formatted information of the request are examined and compared with the contents of the configuration file 404 to determine an appropriate user-mode process that should service the request. (*See page 14, lines 8-11.*)

The implementation of the present Application states the extension-identifying portion of the request (uniform resource locator (URL)) is not used in making the determination as to which user-mode process should service the request. (*See page 14, lines 5-8 of the present Application.*) An extension in a URL is located at the very end of the URL. For example, the extension for the

1 URL `http://www.anyname.com/index.html` is “.html”. Again, such extensions are
2 *not* used by the claimed invention.

3 The hierarchical formatted information of a request will generally be in the
4 form of `http://foo.com/fooapp4/.../....`. More specifically, “foo.com” is one portion
5 of the hierarchical information, “fooapp4” is another portion of the hierarchical,
6 and so on. This is only one example of the form the hierarchical information may
7 take. When the request, such as the URL shown, is received, the UL service 402
8 will compare it with the contents of the configuration file 404. (*See page 16, lines*
9 *8-10; and Figs. 4-5 of the present Application.*) Here, a first portion of
10 hierarchical information is “foo.com.” This first portion of the request is found in
11 a Config Group A, which is illustrated in Fig. 5. Next, “fooapp4” is compared
12 with the contents of the configuration file 404. This subsequent portion “fooapp4”
13 is not referenced in one of the Config Groups, so a user-mode process is selected
14 from an application pool that corresponds to Config Group A. (*See page 8-14; and*
15 *Fig. 5.*)

16 Turning now to the teachings relied upon the Office to reject the claims. The
17 *AAPA* describes a web server process (*user-mode only*) that receives a request that
18 includes a URL. The web server process accesses a mapping function to identify
19 an appropriate worker process based on the extension-identifying portion of the
20 URL in the request. (*Page 11, line 24 – page 12, line 16 of the present*
21 *Application.*) The extension-identifying portion of the URL is not hierarchical
22 information.

23 The *AAPA* does not describe a system in which a the hierarchical identifier
24 is used to identify an appropriate user-mode process for handling the request. The
25 *AAPA* does not have any functionality that would allow it to handle any data that

1 is associated with hierarchical data. Moreover, the *AAPA* does not describe
2 functionality that uses kernel-mode processes to identify user-mode processes
3 based on hierarchical identifiers.

4 The Office recognizes the deficiencies of the *AAPA*. (*See Point 9 of the*
5 *Office Action dated June 13, 2005.*) In order to make up for those deficiencies, the
6 Office has relied upon the teachings of *Benitez*. The Office asserts the
7 combination of the *AAPA* and *Benitez* render the claims of the present Application
8 unpatentable. In the following, Appellant discusses the *Benitez* patent. The
9 discussion will show that *Benitez* does not make up of the deficiencies that the
10 Office acquiesces are present in the *AAPA*. Therefore, the combination of *AAPA*
11 in view of *Benitez* is unable to substantiate a proper rejection under 35 U.S.C.
12 103(a).

13 *Benitez* is directed to a system that partitions an application program into
14 page segments by observing the manner in which the application program is
15 conventionally installed. A minimal portion of the application program is installed
16 on a client system, yet the user launches the application in the same way that
17 applications installed locally on other client file systems are started. (*Benitez*;
18 *Abstract.*)

19 *Benitez* describes the use of a client streaming software 604 to implement
20 the system. (*See paragraph [0187].*) The client software [604] includes a kernel-
21 mode streaming file system driver [905] and a user-mode client [902] to
22 implement the system. (*See paragraph [0193] of Benitez.*) *Benitez* further
23 describes mapping certain filenames to indicate that those files, which would
24 typically be installed on the client device, are actually available from a server.
25 The user-mode client 902 is designed to act as an intermediary between the

1 streaming file system driver 905 and the server when a file is retrieved from the
2 server. The server uses an application streaming server 903 to deliver data back to
3 the user-mode client 902. (See paragraph [0193], page 11, lines 1-4.)

4 Paragraph [0161] of *Benitez* describes how files that would normally be
5 local on a user's system are stored on a server for retrieval when their use is
6 desired. To achieve this result, *Benitez* uses conventional file spoofing. As is
7 described in the paragraph, a location of the file or files that would normally be on
8 the user's system is added to a spoofing database. The location reference in the
9 database includes a mapping to a location on the server where the file is actually
10 located. When the file system on the user's system requires a spoofed file, the
11 database is referenced as to where the file is located and then retrieved. In the
12 *Benitez* system, the client software 604 (i.e., the streaming file system driver 905)
13 performs this retrieval function.

14 The Office contends the system of *Benitez* makes up for the deficiencies of
15 the *AAPA*, and that the combined teachings of the *AAPA* and *Benitez* render the
16 claims of the present Application unpatentable.

17 The described system according to *Benitez* uses a simple database to store
18 references to spoofed files. When a file is called by a user's client system, and the
19 file is not local, the client software 604 simply uses the file name to retrieve a
20 reference to where the file is located on the server. (See paragraph [161] of
21 *Benitez*.) Whether or not the file name is hierarchical in nature makes absolutely
22 no difference to the *Benitez* system; the *Benitez* system simply searches for the file
23 name and the process ends at the point the filename is found, or not. As a matter
24 of fact, the *Benitez* makes no reference to using hierarchical identifiers.
25

1 Accordingly, the combination of *AAPA* and *Benitez* fails to disclose or
2 suggest at least “comparing the hierarchical identifier with at least a portion of a
3 configuration file to identify an appropriate user-mode process for handling the
4 request,” as understood in the context of the Appellant’s disclosure.

5 For the reasons given above, the combination of *Benitez* and *AAPA* does
6 not teach or suggest the subject matter of claim 1. Hence, for at least this reason,
7 this claim is allowable.

8
9 **Claim 3 and 4**

10 Claim 3 depends indirectly from claim 1 and recites that generating the
11 configuration file (recited in claim 2) comprises: defining one or more logical
12 associations between at least one candidate hierarchical identifier and at least one
13 candidate user-mode process; and maintaining the one or more logical associations
14 in a configuration store.

15 On page 7 of the Final Office Action dated June 13, 2005, Point 19, the
16 Office argues that claim 3 has been “rejected for the same reason as claims 13-
17 24.” Appellant disagrees and traverses this rejection.

18 The Office, on page 4, Point 11, states “Benitez discloses a spoof database
19 being generated by a kernel-mode process (i.e. a client file spoofer), where
20 access/request to a matching entry in the spoof database is directed by the client
21 file spoofer to a remote system to handle the request [Benitez, pg. 8, lines 12-20].”
22 The Office makes this reference to *Benitez* in the text rejecting claims 14-16. The
23 Appellant believes this text applies to the rejection of claim 3 as well. But the
24 Appellant cannot be positive, as the Office has not directly cited portions of *AAPA*
25 and *Benitez* that are believed to render claim 3 obvious.

1 The Appellant has carefully considered the subject matter of the
2 combination of *AAPA* and *Benitez* and finds nothing that discloses or suggests
3 “defining one or more logical associations between at least one candidate
4 hierarchical identifier and at least one candidate user-mode process; and
5 maintaining the one or more logical associations in a configuration store,” as is
6 understood in the context of the Appellant’s disclosure. Hence, for at least this
7 reason, this claim is allowable.

8
9 **Claims 13 and 17-24**

10 Claim 13 recites a computer-readable medium having computer-executable
11 instructions for performing steps comprising:

- 12 • causing a kernel-mode process in a server device to compare a
13 hierarchical identifier associated with a client device generated
14 request with at least a portion of a configuration file to identify a
15 most applicable user-mode process for handling the client device
16 generated request within the server device; and
17 • causing the kernel-mode process to provide the client device
18 generated request to the identified most applicable user-mode
19 process.

20 Again, an implementation described in the present Application teaches
21 receiving a request at a *kernel-mode* universal listener (UL) service 402 and
22 comparing the request with content of a *kernel-mode* configuration file 404. (*See*
23 *page 14, lines 3-5; and Fig. 4 of the present Application.*) Hierarchical formatted
24 information of the request are examined and compared with the contents of the
25 configuration file 404 to determine an appropriate user-mode process that should
service the request. (*See page 14, lines 8-11.*)

1 The implementation of the present Application states the extension-
2 identifying portion of the request (URL) is not used in making the determination
3 as to which user-mode process should service the request. (*See page 14, lines 5-8*
4 *of the present Application.*) An extension in a URL is located at the very end of
5 the URL. For example, the extension for the URL
6 http://www.anyname.com/index.html is “.html”. Again, such extensions are *not*
7 used by the claimed invention.

8 The hierarchical formatted information of a request will generally be in the
9 form of http://foo.com/fooapp4/.../.... More specifically, “foo.com” is one portion
10 of the hierarchical information, “fooapp4” is another portion of the hierarchical,
11 and so on. This is only one example of the form the hierarchical information may
12 take. When the request, such as the URL shown, is received, the UL service 402
13 will compare it with the contents of the configuration file 404. (*See page 16, lines*
14 *8-10; and Figs. 4-5 of the present Application.*) Here, a first portion of
15 hierarchical information is “foo.com.” This first portion of the request is found in
16 a Config Group A, which is illustrated in Fig. 5. Next, “fooapp4” is compared
17 with the contents of the configuration file 404. This subsequent portion “fooapp4”
18 is not referenced in one of the Config Groups, so a user-mode process is selected
19 from an application pool that corresponds to Config Group A. (*See page 8-14; and*
20 *Fig. 5.*)

21 Turing now to the teachings relied upon the Office to reject the claims. The
22 *AAPA* describes a web server process (*user-mode only*) that receives a request that
23 includes a URL. The web server process accesses a mapping function to identify
24 an appropriate worker process based on the extension-identifying portion of the
25 URL in the request. (*Page 11, line 24 – page 12, line 16 of the present*

1 Application.) The extension-identifying portion of the URL is not hierarchical
2 information.

3 The *AAPA* does not describe a system in which a hierarchical identifier is
4 used to identify an appropriate user-mode process for handling the request. The
5 *AAPA* does not have any functionality that would allow it to handle any data that
6 is associated with hierarchical data. Moreover, the *AAPA* does not describe
7 functionality that uses kernel-mode processes to identify user-mode processes
8 based on hierarchical identifiers.

9 The Office recognizes the deficiencies of the *AAPA*. (See *Point 9 of the*
10 *Office Action dated June 13, 2005*.) In order to make up for those deficiencies, the
11 Office has relied upon the teachings of *Benitez*. The Office asserts the
12 combination of the *AAPA* and *Benitez* render the claims of the present Application
13 unpatentable. In the following, Appellant discusses the *Benitez* patent. The
14 discussion will show that *Benitez* does not make up of the deficiencies that the
15 Office acquiesces are present in the *AAPA*. Therefore, the combination of *AAPA*
16 in view of *Benitez* is unable to substantiate a proper rejection under 35 U.S.C.
17 103(a).

18 *Benitez* is directed to a system that partitions an application program into
19 page segments by observing the manner in which the application program is
20 conventionally installed. A minimal portion of the application program is installed
21 on a client system, yet the user launches the application in the same way that
22 applications installed locally on other client file systems are started. (*Benitez*;
23 *Abstract*.)

24 *Benitez* describes the use of a client streaming software 604 to implement
25 the system. (See *paragraph [0187] of Benitez*.) The client software [604] includes

1 a kernel-mode streaming file system driver [905] and a user-mode client [902] to
2 implement the system. (See paragraph [0193].) Benitez further describes
3 mapping certain filenames to indicate that those files, which would typically be
4 installed on the client device, are actually available from a server. The user-mode
5 client 902 is designed to act as an intermediary between the streaming file system
6 driver 905 and the server when a file is retrieved from the server. The server uses
7 an application streaming server 903 to deliver data back to the user-mode client
8 902. (See paragraph [0193], page 11, lines 1-4.)

9 Paragraph [0161] of Benitez describes how files that would normally be
10 local on a user's system are stored on a server for retrieval when their use is
11 desired. To achieve this result, Benitez uses conventional file spoofing. As is
12 described in the paragraph, a location of the file or files that would normally be on
13 the user's system is added to a spoofing database. The location reference in the
14 database includes a mapping to a location on the server where the file is actually
15 located. When the file system on the user's system requires a spoofed file, the
16 database is referenced as to where the file is located and then retrieved. In the
17 Benitez system, the client software 604 (i.e., the streaming file system driver 905)
18 performs this retrieval function.

19 The Office contends the system of Benitez makes up for the deficiencies of
20 the AAPA, and that the combined teachings of the AAPA and Benitez render the
21 claims of the present Application unpatentable.

22 The described system according to Benitez uses a simple database to store
23 references to spoofed files. When a file is called by a user's client system, and the
24 file is not local, the client software 604 simply uses the file name to retrieve a
25 reference to where the file is located on the server. (See paragraph [161].)

1 Whether or not the file name is hierarchical in nature makes absolutely no
2 difference to the *Benitez* system; the *Benitez* system simply searches for the file
3 name and the process ends at the point the filename is found, or not. As a matter
4 of fact, the *Benitez* makes no reference to using hierarchical identifiers.

5 Accordingly, the combination of *AAPA* and *Benitez* fails to disclose or
6 suggest at least the subject matter “compare a hierarchical identifier associated
7 with a client device generated request with at least a portion of a configuration file
8 to identify a most applicable user-mode process for handling the client device
9 generated request within the server device,” as understood in the context of the
10 Appellant’s disclosure.

11 In addition, the Office contends, on page 4 of the Final Office Action dated
12 June 13, 2005, Point 9, that “*Benitez* disclosed that requests made to an
13 application streaming file system comprising the comparison of hierarchical
14 identifier to a configuration file in a kernel-mode environment.” The Applicant
15 submits that *Benitez* discloses many client based operational elements. Some of
16 those elements are part of a kernel space 617 (*see figure 6A*) and some of those
17 elements are part of a user space 618 (*again, see figure 6A*). These spaces
18 generally interface with an application streaming server 803, 903, 1005, or 1103
19 that is part of a server machine (*see figures 8, 9, 10 and 11*).

20 So, what *Benitez* discloses is the manner in which elements of a client
21 device interfaces with a server machine. However, contrary to the Office’s
22 contention, *Benitez* does not disclose or suggest how the server machine is
23 implemented. That is, nothing in *Benitez* discusses the deployment of various
24 server related operational elements in a user space and/or a kernel space. Thus,
25 even if one of ordinary skill in the art were to combine *Benitez* with *AAPA*, the

1 limitation of “causing a kernel-mode process in a server...,” as is recited in claim
2 13, would not be met.

3 For the reasons given above, the combination of *Benitez* and *AAPA* does
4 not teach or suggest the subject matter of claim 13. Hence, for at least this reason,
5 this claim is allowable.

6 7 **Claim 14**

8 Claim 14 depends directly from claim 13 and recites causing a user-mode
9 administrative process to generate the configuration file.

10 In rejecting the claim, the Office, on page 4 of the Final Office Action
11 dated June 13, 2005, Point 11, states “Benitez discloses a spoof database being
12 generated by a kernel-mode process (i.e. a client file spoofer), where
13 access/request to a matching entry in the spoof database is directed by the client
14 file spoofer to a remote system to handle the request [Benitez, pg. 8, lines 12-20].”
15 Appellant disagrees and traverses this rejection.

16 The Appellant has carefully considered the subject matter of the
17 combination of *AAPA* and *Benitez* and finds nothing that discloses or suggests
18 “causing a user-mode administrative process to generate the configuration file,” as
19 is understood in the context of the Appellant’s disclosure. In particular, the
20 combination of *AAPA* and *Benitez* makes no reference to the generating a spoof
21 database using a kernel-mode process, as is suggested by the Office. Therefore,
22 the combination does not disclose or suggest the subject matter of the claim.
23 Hence, for at least this reason, this claim is allowable.

1 **Claims 15 and 16**

2 Claim 15 depends indirectly from claim 13 and recites wherein causing the
3 user-mode administrative process to generate the configuration file, further
4 includes: providing a configuration store suitable for access by the user-mode
5 administrative process, wherein the configuration store defines one or more logical
6 associations between at least one candidate hierarchical identifier and at least one
7 candidate user-mode process.

8 In rejecting the claim, the Office, on page 4 of the Final Office Action
9 dated June 13, 2005, Point 11, states “Benitez discloses a spoof database being
10 generated by a kernel-mode process (i.e. a client file spoofer), where
11 access/request to a matching entry in the spoof database is directed by the client
12 file spoofer to a remote system to handle the request [Benitez, pg. 8, lines 12-20].”
13 Appellant disagrees and traverses this rejection.

14 The Appellant has carefully considered the subject matter of the
15 combination of *AAPA* and *Benitez* and finds nothing that discloses or suggests
16 “providing a configuration store suitable for access by the user-mode
17 administrative process, wherein the configuration store defines one or more logical
18 associations between at least one candidate hierarchical identifier and at least one
19 candidate user-mode process,” as is understood in the context of the Appellant’s
20 disclosure. In particular, the combination of *AAPA* and *Benitez* makes no
21 reference to the generating a spoof database using a kernel-mode process, as is
22 suggested by the Office. Therefore, the combination does not disclose or suggest
23 the subject matter of the claim. Hence, for at least this reason, this claim is
24 allowable.
25

(C) Claims 1-11 and 13-24 are allowable because the Office has used impermissible hindsight as motivation to combine *AAPA* and *Benitez*.

Claims 1-11 and 13-24

In combining the *AAPA* and *Benitez*, the office states a “person with ordinary skill in the art would have been motivated to implement the powerful/efficient kernel-mode approach over the less efficient user-mode approach to address the issue of *AAPA*.” (See page 4 of the Final Office Action, Point 10.) Appellant respectfully submits, for the reasons that follow, that the Office’s reasoning for combining the *AAPA* and *Benitez* is based on *hindsight* knowledge and therefore flawed. As is stated in *ATD Corporation v. Lydall, Inc.*, 48 USPQ 2d 1321, 1329 (Fed. Cir. 1998), the “determination of obviousness can not be based on the hindsight combination of components selectively culled from the prior art to fit the parameters of the patented invention.”

The *AAPA* describes a web server process that receives a request that includes a URL. The web server process accesses a mapping function to identify an appropriate worker process based on the extension-identifying portion of the URL in the request. (Page 11, line 24 – page 12, line 16 of the present Application.) The extension-identifying portion of the URL is not hierarchical information, and the request is handled exclusively on the user-mode side, as is delineated by a dashed line of Fig. 2 of the present Application.

Benitez is directed to a system that partitions an application program into page segments by observing the manner in which the application program is

1 conventionally installed. A minimal portion of the application program is installed
2 on a client system, yet the user launches the application in the same way that fully
3 installed applications on other client file systems are started. (*See Benitez;*
4 *Abstract.*) The Office states the *Benitez* system uses a “pure kernel-mode” process
5 for handling requests.

6 When stating that *AAPA* and *Benitez* may be combined, the Office fails to
7 address and explain why one of ordinary skill in the art would look to a system
8 that makes use of file spoofing, as disclosed by *Benitez*, when attempting to rectify
9 the deficiencies of the *AAPA*. File spoofing is a technique used to map files that
10 would normally be stored on a local client device to another device, such as a
11 server. (*See page 8, Benitez paragraph [0161].*) *Benitez* does not discuss, suggest,
12 or address the problems associated with handling web server requests, the use of
13 web server processes, or the use of file extensions to identify a web server process
14 for handling such web server requests. These are exactly the concerns and needs
15 expressed in conjunction with *AAPA*. (*See page 2-3 of the present Application.*)

16 More succinctly put, file spoofing and the handling of web server requests
17 have nothing in common. Moreover, as is described in *Benitez*, the file spoofing
18 occurs at the client side, not the server side where the claimed invention normally
19 operates. (*See pages 7 and 8, Benitez paragraphs [0158] – [0161].*)

20 In addition, the Office’s statement that the *Benitez* system uses a “pure
21 kernel-mode” process is incorrect. *Benitez* discloses many client based
22 operational elements. Some of those elements are part of a kernel space 617 (*see*
23 *figure 6A*) and some of those elements are part of a user space 618 (*again, see*
24 *figure 6A*). These spaces generally interface with an application streaming server
25 803, 903, 1005, or 1103 that is part of a server machine (*see figures 8, 9, 10 and*

1 11). Whether the server machine is configured to operate in a "pure kernel-mode"
2 configuration is never mentioned in *Benitez*, and the clients described in *Benitez*
3 are designed to use both user and kernel modes. (See *Benitez*, paragraph [0193].)
4 Actually, although *Benitez* indicates that all kernel-mode designs have some
5 advantages, the reference steers away from such a design in favor of using the
6 combination of both kernel and user spaces on the client devices. Therefore,
7 further discussion of kernel-mode implementations is not provided by *Benitez*.

8 Based on the foregoing, it is apparent the Office used hindsight knowledge
9 gleaned from the present claimed invention when it decided to combine *AAPA* and
10 *Benitez*. First, the technical problem solved by *AAPA* (i.e., identifying web server
11 processes) is completely unrelated to the technical problem solved by *Benitez* (i.e.,
12 file spoofing). Second, the motivation given by the Office, which was supposedly
13 disclosed by *Benitez*, is not supported by the patent document.

14 Appellant has provided compelling evidence that the Office used *hindsight*
15 knowledge gleaned from the present Application when the current § 103 rejection
16 was formulated. Accordingly, in addition to the technical deficiencies of the
17 *AAPA* and *Benitez*, Appellant respectfully submits the current 35 U.S.C. 103(a)
18 rejection is improperly formulated. Reconsideration and withdrawal of the
19 rejection are requested.
20
21
22
23
24
25

(D) Claims 1-11 and 13-24 are allowable because *AAPA* and *Benitez* are from non-analogous arts

Claims 1-11 and 13-24

A combination of elements from non-analogous sources, in a manner that reconstructs the applicant's invention only with the benefit of hindsight, is insufficient to present a prima facie case of obviousness. *In re Oetiker*, 24 USPQ 2d 1443, 1446 (Fed. Cir. 1992). Appellant submits that *Benitez* is from non-analogous art for the following reasons.

The Office has admitted, on the Record, that *Benitez* is non-analogous art when compared against *AAPA* and the claimed invention. (*See page 8, Point 25, Office Action dated June 13, 2006.*) The determination that a reference is from non-analogous art involves a two-step process. First, we decide if the reference is within the field of the inventor's endeavor. If it is not, we proceed to determine whether the reference is reasonably pertinent to the particular problem with which the inventor was involved. *In re Wood*, 202 USPQ 171, 174 (C.C.P.A. 1979). The Office has already acquiesced that *Benitez* is from non-analogous art. Nonetheless, the Appellant provides the following support for the Office's conclusion.

As was discussed earlier in this Brief, *AAPA*, which is from an analogous art as the present claimed invention, describes a web server process that receives a request that includes a URL. The web server process accesses a mapping function to identify an appropriate worker process based on the extension-identifying portion of the URL in the request. (*Page 11, line 24 – page 12, line 16 of the*

1 present Application.) The extension-identifying portion of the URL is not
2 hierarchical information, and the request is handled exclusively on the user-mode
3 side, as is delineated by a dashed line of Fig. 2 of the present Application.

4 Benitez is directed to a system that partitions an application program into
5 page segments by observing the manner in which the application program is
6 conventionally installed. A minimal portion of the application program is installed
7 on a client system, yet the user launches the application in the same way that fully
8 installed applications on other client file systems are started. (See Benitez;
9 Abstract.) Furthermore, Benitez discloses a file spoofing technique used to map
10 files that would normally be stored on a local client device to another device, such
11 as a server. (See page 8, Benitez paragraph [0161].)

12 Similar to the present claimed invention, AAPA is in the field related to
13 handling requests received by a web server and identifying a process to handle
14 such requests. In contrast, Benitez is in the field that relates to file spoofing at the
15 client side. These two fields are not related. Therefore, it follows that the present
16 claimed invention is from a technical field that is different than the field associated
17 with Benitez. The Board is reminded, the mere fact that AAPA and Benitez have
18 operations that require computers is not enough to determine that the technologies
19 are from analogous arts. See *Wang Labs., Inc. v. Toshiba Corp.*, 26 USPQ 2d
20 1767, 1773 (Fed. Cir. 1993). This case articulated that the relied upon art was not
21 “in the same field of endeavor as the claimed subject matter merely because it
22 relates to memories.” The foregoing satisfies the first step in the two-step process
23 articulated above.

24 Benitez does not discuss, suggest, or address the problems associated with
25 handling web server requests, the use of web server processes, or the use of file

1 extensions or hierarchical identifiers to identify a web server process for handling
2 such web server requests. These are exactly the concerns and needs expressed in
3 conjunction with *AAPA* (using extensions) and the present claimed invention
4 (using hierarchical identifiers). That is, the present claimed invention is also
5 concerned with identifying web server processes for web server requests, but does
6 so in a different manner than contemplated by *AAPA*. Therefore, one can not
7 reasonably say that *Benitez* is “reasonably pertinent to the particular problem with
8 which the inventor was involved.” The foregoing satisfies the second step in the
9 two-step process articulated above.

10 In accordance with the above, Appellant believes the Office used a
11 reference from non-analogous art (*Benitez*) when the current § 103 rejection was
12 formulated. Accordingly, in addition to the technical deficiencies of the *AAPA* and
13 *Benitez* and the Office’s impermissible use of *hindsight* knowledge, Appellant
14 respectfully submits the current 35 U.S.C. 103(a) rejection is improperly
15 formulated. Reconsideration and withdrawal are requested.
16
17
18
19
20
21
22
23
24
25

1
2 **Conclusion**

3 Claims 1-11 and 13-24 are in condition for allowance. Appellant
4 respectfully requests reconsideration and prompt allowance of the subject
5 application.
6
7
8

9
10 Date: 3-13-2006

Respectfully Submitted,

By: 

Tim R. Wyckoff
Lee & Hayes, pllc
Reg. No. 46,175
(206) 315-4001 ext. 110

(viii) Claims Appendix

1
2 1. (Previously amended) A method comprising:
3 receiving a request from a client device, the request comprising a
4 hierarchical identifier;
5 comparing the hierarchical identifier with at least a portion of a
6 configuration file to identify an appropriate user-mode process for handling the
7 request; and
8 providing the request to the identified appropriate user-mode process.
9

10
11 2. (Previously amended) The method as recited in Claim 1, further
12 comprising:
13 generating the configuration file via a user-mode administrative process.
14

15
16 3. (Previously amended) The method as recited in Claim 2, wherein
17 generating the configuration file comprises:
18 defining one or more logical associations between at least one candidate
19 hierarchical identifier and at least one candidate user-mode process; and
20 maintaining the one or more logical associations in a configuration store.
21
22
23
24
25

1 4. (Previously amended) The method as recited in Claim 3, further
2 comprising:

3 maintaining one or more logical rules suitable for use in identifying the
4 appropriate user-mode process for handling the request.
5

6 5. (Previously amended) The method as recited in Claim 1, wherein
7 providing the request to the identified appropriate user-mode process further
8 comprises:
9

10 providing the request via a non-shared interface associated with the
11 identified appropriate user-mode process.
12

13 6. (Previously amended) The method as recited in Claim 1, further
14 comprising selectively queuing the request prior to providing the request to the
15 identified most applicable user-mode process.
16

17
18 7. (Previously amended) The method as recited in Claim 1, wherein
19 the request comprises a uniform resource locator (URL).
20

21 8. (Previously amended) The method as recited in Claim 1, wherein
22 the appropriate user-mode process includes a user-mode web server process.
23
24
25

1 9. (Previously amended) The method as recited in Claim 1, wherein
2 the appropriate user-mode process comprises at least one user-mode worker
3 process.
4

5 10. (Previously amended) The method as recited in Claim 1, further
6 comprising:
7

8 receiving the client request using a kernel-mode communication protocol
9 process; and
10

11 providing the request to a kernel-mode process.
12

13 11. (Previously amended) The method as recited in Claim 10, wherein
14 the kernel-mode communication protocol process comprises a kernel-mode
15 TCP/IP process.
16
17
18
19
20
21
22
23
24
25

12. (Canceled)

13. (Previously amended) A computer-readable medium having computer-executable instructions for performing steps comprising:

causing a kernel-mode process in a server device to compare a hierarchical identifier associated with a client device generated request with at least a portion of a configuration file to identify a most applicable user-mode process for handling the client device generated request within the server device; and

causing the kernel-mode process to provide the client device generated request to the identified most applicable user-mode process.

14. (Original) The computer-readable medium as recited in Claim 13, having further computer-executable instructions for performing steps comprising:

causing a user-mode administrative process to generate the configuration file.

1 15. (Original) The computer-readable medium as recited in Claim 14,
2 wherein causing the user-mode administrative process to generate the
3 configuration file, further includes:

4 providing a configuration store suitable for access by the user-mode
5 administrative process, wherein the configuration store defines one or more logical
6 associations between at least one candidate hierarchical identifier and at least one
7 candidate user-mode process.
8

9
10 16. (Previously amended) The computer-readable medium as recited in
11 Claim 15, wherein the configuration store further includes one or more logical
12 rules suitable for use by the kernel-mode process in identifying the most
13 applicable user-mode process for handling the client device generated request
14 within the server device.
15
16
17
18
19
20
21
22
23
24
25

1 17. (Previously amended) The computer-readable medium as recited in
2 Claim 13, wherein causing the kernel-mode process to provide the client device
3 generated request to the identified most applicable user-mode process further
4 includes:

5 providing a non-shared interface between the kernel-mode process and the
6 identified most applicable user-mode process, such that the client device generated
7 request can be provided to the identified most applicable user-mode process via
8 the non-shared interface.
9

10
11 18. (Previously amended) The computer-readable medium as recited in
12 Claim 13, wherein causing the kernel-mode process to provide the client device
13 generated request to the identified most applicable user-mode process further
14 includes:

15 selectively queuing the client device generated request prior to providing
16 the request to the identified most applicable user-mode process.
17

18
19 19. (Original) The computer-readable medium as recited in Claim 13,
20 wherein the hierarchical identifier includes a uniform resource locator (URL).
21
22
23
24
25

1 20. (Original) The computer-readable medium as recited in Claim 13,
2 wherein the most applicable user-mode process includes a user-mode web server
3 process.

4
5 21. (Original) The computer-readable medium as recited in Claim 13,
6 wherein the most applicable user-mode process includes at least one user-mode
7 worker process.

8
9
10 22. (Previously amended) The computer-readable medium as recited in
11 Claim 13, having further computer-executable instructions for performing steps
12 comprising:

13 receiving the client device generated request using a kernel-mode
14 communication protocol process; and

15 providing the client device generated request to the kernel-mode process.
16

17
18 23. (Original) The computer-readable medium as recited in Claim 22,
19 wherein the kernel-mode communication protocol process includes a kernel-mode
20 TCP/IP process.
21
22
23
24
25

1 24. (Previously amended) The computer-readable medium as recited in
2 Claim 13, having further computer-executable instructions for performing steps
3 comprising:

4 causing the identified most applicable user-mode process to handle the
5 client device generated request.

6
7 Claims 25 - 33. (Canceled)
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

1 (ix) Evidence Appendix

2 None.

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

(x) Related Proceedings Appendix

None.